INTAREA WG
Internet-Draft
Intended status: Informational
Expires: September 7, 2012

M. Boucadair
France Telecom
J. Touch
USC/ISI
P. Levis
France Telecom
R. Penno
Juniper Networks
March 06, 2012

Analysis of Solution Candidates to Reveal a Host Identifier in Shared Address Deployments
draft-ietf-intarea-nat-reveal-analysis-01

Abstract

This document analyzes a set of solution candidates to mitigate some of the issues encountered when address sharing is used. In particular, this document focuses on means to reveal a host identifier (HOST_ID) when a Carrier Grade NAT (CGN) or application proxies are involved in the path. This host identifier must be unique to each host under the same shared IP address.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 7, 2012.

Copyright Notice

Table of Contents

1. Introduction

As reported in [RFC6269], several issues are encountered when an IP address is shared among several subscribers. Examples of such issues are listed below:

- Implicit identification (Section 13.2 of [RFC6269])
- SPAM (Section 13.3 of [RFC6269])
- Blacklisting a mis-behaving user (Section 13.1 of [RFC6269])
- Redirect users with infected machines to a dedicated portal (Section 5.1 of [RFC6269])

The sole use of the IPv4 address is not sufficient to uniquely distinguish a host. As a mitigation, it is tempting to investigate means which would help in disclosing an information to be used by the remote server as a means to uniquely disambiguate packets of hosts using the same IPv4 address.

The risk of not mitigating these issues are: OPEX increase for IP connectivity service providers (costs induced by calls to a hotline), revenue loss for content providers (loss of users audience), customers unsatisfaction (low quality of experience, service segregation, etc.).

1.1. Problem to Be Solved

Observation: Today, some servers use the source IPv4 address as an identifier to treat some incoming connections differently. Tomorrow, due to the introduction of CGNs (e.g., NAT44 [I-D.ietf-behave-lsn-requirements], NAT64 [RFC6146]), that address will be shared. In particular, when a server receives packets from the same source address, because this address is shared, the server does not know which host is the sending host.

Objective: The server should be able to sort out the packets by sending host.

Requirement: The server must have extra information than the source IP address to differentiate the sending host. We call HOST_ID this information.

For all solutions analyzed, we provide answers to the following questions:

What is the HOST_ID? It must be unique to each host under the same IP address. It does not need to be globally unique. Of course, the combination of the (public) IP source address and the identifier (i.e., HOST_ID) ends up being relatively unique. As unique as today's 32-bit IPv4 addresses which, today, can change

when a host re-connects.

Where is the HOST_ID? (which protocol, which field): If the HOST_ID is put at the IP level, all packets will have to bear the identifier. If it is put at a higher connection-oriented level, the identifier is only needed once in the session establishment phase (for instance TCP three-way-handshake), then, all packets received in this session will be attributed to the HOST_ID designated during the session opening.

Who puts the HOST_ID? For almost all the analyzed solutions, the address sharing function injects the HOST_ID. When there are several address sharing functions in the data path, we describe to what extent the proposed solution is efficient. Another option to avoid potential performance degradation is to let the host inject its HOST_ID but the address sharing function will check its content (just like an IP anti-spoofing function).

What are the security considerations? Security considerations are common to all analyzed solutions (see Section 6). Privacy-related aspect are discussed in Section 4.

1.2. IPv6 May Also Be Concerned

Some of the issues mentioned in Section 1.1 are independent of IPv4 vs. IPv6. Even in IPv6, address sharing can be used for a variety of reasons (e.g., to hide network topology, to defeat hosts from offering network services directly, etc.).

A solution to reveal HOST_ID is also needed in IPv6 deployment.

1.3. Purpose and Scope

The purpose of this document is not to argue in favor of mandating the use of a HOST_ID but to identify encountered issues, proposed solutions and their limitations.

The purpose of this document is to analyze a set of solution candidates and to assess to what extent they solve the problem (see Section 1.1). Below are listed the solutions analyzed in the document:

- Use the Identification field of IP header (denoted as IP-ID, Section 3.1).
- Define a new IP option (Section 3.2).
- Assign port sets (Section 3.3).

- Use ICMP (Section 3.3).
- Define a new TCP Option (Section 3.5).
- Enable Proxy Protocol ((Section 3.6)).
- Activate HIP (Section 3.7).
- Inject application headers (Section 3.8).

2. Synthesis

The following Table 1 summarizes the approaches analyzed in this document.

- "Success ratio" indicates the ratio of successful communications when the option is used. Provided figures are inspired from the results documented in [Options].
- "Deployable today" indicates if the solution can be generalized without any constraint on current architectures and practices.
- "Possible Perf Impact" indicates the level of expected performance degradation. The rationale behind the indicated potential performance degradation is whether the injection requires some treatment at the IP level or not.
- "OS TCP/IP Modif" indicates whether a modification of the OS TCP/IP stack is required at the server side.

	IP Option	TCP Option	IP-ID	HTTP Header (XFF)	Proxy	Port Set	HIP	ICMP
UDP	Yes	No	Yes	No	No	Yes		Yes
TCP	Yes	Yes	Yes	No	Yes	Yes		Yes
HTTP	Yes	Yes	Yes	Yes	Yes	Yes		Yes
Encrypted Traffic	Yes	Yes	Yes	No	Yes	Yes		Yes
Success Ratio	30%	99%	100%	100%	Low	100%	Low	~100% (6)
Possible Perf Impact	High	Med to High	Low to Med	Med to High	High	No	N/A	High
OS TCP/IP Modif	Yes	Yes	Yes	No	No	No		Yes
Deployable Today	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Notes			(1)	(2)		(1) (3)	(4) (5)	(7)

Notes:

(1) Requires mechanism to advertise NAT is participating in this scheme (e.g., DNS PTR record).
(2) This solution is widely deployed.
(3) When the port set is not advertised, the solution is less efficient for third-party services.
(4) Requires the client and the server to be HIP-compliant and HIP infrastructure to be deployed.
(5) If the client and the server are HIP-enabled, the address sharing function does not need to insert a host-hint. If the client is not HIP-enabled, designing the device that performs address sharing to act as a UDP/TCP-HIP relay is not viable.
(6) Implementation specific.
(7) The solution is inefficient is various scenarios as discussed in Section 3.

Figure 1: Table 1: Summary of analyzed solutions.

According to the above table and the analysis elaborated in Section 3:

- IP Option, IP-ID and Proxy Protocol proposals are broken;
- HIP is not largely deployed;
- The use of Port Set may contradict the port randomization [RFC6056] requirement identified in [RFC6269]. This solution can be used by a service provider for the delivery of its own service offerings relying on implicit identification.
- XFF is de facto standard deployed and supported in operational networks (e.g., HTTP Severs, Load-Balancers, etc.).
- From an application standpoint, the TCP Option is superior to XFF since it is not restricted to HTTP. Nevertheless XFF is compatible with the presence of address sharing and load-balancers in the communication path. To provide a similar functionality, the TCP Option may be extended to allow conveying a list of IP addresses and port numbers to not lose the source IP address in the presence of load-balancers. Another alternative is to combine the usage of both the HOST_ID TCP Option and XFF. Note that TCP Option requires the modification of the OS TCP/IP stack of remote servers; which can be seen as a blocking point.

For all HOST_ID proposals, the following recommendations are made:

Uniqueness of identifiers in HOST_ID: It is RECOMMENDED that HOST_IDs be limited to providing local uniqueness rather than global uniqueness.

Refresh rate of HOST_ID: Address sharing function SHOULD NOT use permanent HOST_ID values.

Manipulate HOST_IDs: Address sharing function SHOULD be able to strip, re-write and add HOST_ID fields.

Interference between HOST_IDs: An address sharing function, able to inject HOST_IDs in several layers, SHOULD reveal subsets of the same information (e.g., full IP address, lower 16 bits of IP address, etc.).

3. Solutions Analysis

3.1. Use the Identification Field of IP Header (IP-ID)

3.1.1. Description

IP-ID (Identification field of IP header) can be used to insert an information which uniquely distinguishes a host among those sharing the same IPv4 address. An address sharing function can re-write the IP-ID field to insert a value unique to the host (16 bits are sufficient to uniquely disambiguate hosts sharing the same IP address). Note that this field is not altered by some NATs; hence some side effects such as counting hosts behind a NAT as reported in [Count].

A variant of this approach relies upon the format of certain packets, such as TCP SYN, where the IP-ID can be modified to contain a 16 bit HOST_ID. Address sharing devices performing this function would require to indicate they are performing this function out of band, possibly using a special DNS record.

3.1.2. Analysis

This usage is not compliant with what is recommended in [I-D.ietf-intarea-ipv4-id-update].

3.2. Define an IP Option

3.2.1. Description

A solution alternative to convey the HOST_ID is to define an IP option [RFC0791]. HOST_ID IP option can be inserted by the address sharing function to uniquely distinguish a host among those sharing the same IP address. An example of such option is documented in [I-D.chen-intarea-v4-uid-header-option]. This IP option allows to convey an IPv4 address, an IPv6 prefix, a GRE key, IPv6 Flow Label, etc.

Another way for using IP option has been described in Section 4.6 of [RFC3022].

3.2.2. Analysis

Unlike the solution presented in Section 3.5, this proposal can apply for any transport protocol. Nevertheless, it is widely known that routers (and other middleboxes) filter IP options. IP packets with IP options can be dropped by some IP nodes. Previous studies demonstrated that "IP Options are not an option" (Refer to

[Not_An_Option], [Options]).

As a conclusion, using an IP option to convey a host-hint is not viable.

3.3. Assign Port Sets

3.3.1. Description

This solution does not require any action from the address sharing function to disclose a host identifier. Instead of assuming all transport ports are associated with one single host, each host under the same external IP address is assigned a restricted port set. These port sets are then advertised to remote servers using off-line means. This announcement is not required for the delivery of internal services (i.e., offered by the service provider deploying the address sharing function) relying on implicit identification.

Port sets assigned to hosts may be static or dynamic.

Port set announcements to remote servers do not require to reveal the identity of individual hosts but only to advertise the enforced policy to generate non-overlapping port sets (e.g., the transport space associated with an IP address is fragmented to contiguous blocks of 2048 port numbers).

3.3.2. Analysis

The solution does not require defining new fields nor options; it is policy-based.

The solution may contradict the port randomization as identified in [RFC6269]. A mitigation would be to avoid assigning static port sets to individual hosts.

The method is convenient for the delivery of services offered by the service provider offering also the IP connectivity service.

3.4. Use ICMP

3.4.1. Description

Another alternative is to convey the HOST_ID using a separate notification channel than the packets issued to invoke the service.

An implementation example is defined in [I-D.yourtchenko-nat-reveal-ping]. This solution relies on a mechanism where the address sharing function encapsulates the

necessary differentiating information into an ICMP Echo Request packet that it sends in parallel with the initial session creation (e.g., SYN). The information included in the ICMP Request Data portion describes the five-tuples as seen on both of the sides of the address sharing function.

3.4.2. Analysis

- This ICMP proposal is valid for both UDP and TCP. Address sharing function may be configurable with the transport protocol which is allowed to trigger those ICMP messages.
- A hint should be provided to the ultimate server (or intermediate nodes) an ICMP Echo Request conveys a HOST_ID. This may be implemented using magic numbers.
- Even if ICMP packets are blocked in the communication path, the user connection does not have to be impacted.
- Some implementations requiring to delay the establishment of a session until receiving the companion ICMP Echo Request, may lead to some user experience degradation.
- Because of the presence of load-balancers in the path, the ultimate server receiving the SYN packet may not be the one which may receive the ICMP message conveying the HOST_ID.
- Because of the presence of load-balancers in the path, the port number assigned by address sharing may be lost. Therefore the mapping information conveyed in the ICMP may not be sufficient to associate a SYN packet with a received ICMP.
- The proposal is not compatible with the presence of cascaded NAT.
- The ICMP proposal will add a traffic overhead for both the server and the address sharing device.
- The ICMP proposal is similar to other mechanisms (e.g., syslog, netflow) for reporting dynamic mappings to a mediation platform (mainly for legal traceability purposes). Performance degradation are likely to be experienced by address sharing functions because ICMP messages are to be sent in particular for each new instantiated mapping (and also even if the mapping exists).
- In some scenarios (e.g., Fixed-Mobile Convergence, Open WiFi, etc.), HOST_ID should be interpreted by intermediate devices which embed Policy Enforcement Points (PEP, [RFC2753]) responsible for granting access to some services. These PEPs need to inspect all received packets in order to find the companion (traffic) messages to be correlated with ICMP messages conveying HOST_IDs. This induces more complexity to these intermediate devices.

3.5. Define a TCP Option

3.5.1. Description

HOST_ID may be conveyed in a dedicated TCP Option. An example is specified in [I-D.wing-nat-reveal-option] which defines a new TCP Option called USER_HINT. This option encloses the TCP client's identifier (e.g., the lower 16 bits of their IPv4 address, their VLAN ID, VRF ID, subscriber ID). The address sharing device inserts this TCP Option into the TCP SYN packet.

3.5.2. Analysis

Using a new TCP Option to convey the HOST_ID does not require any modification to the applications but it is applicable only for TCP-based applications. Applications relying on other transport protocols are therefore left unsolved.

[I-D.wing-nat-reveal-option] discusses the interference with other TCP Options.

The risk related to handling a new TCP Option is low as measured in [Options]. [I-D.abdo-hostid-tcpopt-implementation] provides a detailed implementation and experimentation report of HOST_ID TCP Option. [I-D.abdo-hostid-tcpopt-implementation] investigated in depth the impact of activation HOST_ID in host, address sharing function and the enforcement of policies at the server side. [I-D.abdo-hostid-tcpopt-implementation] reports a failure ratio of 0,103% among top 100000 websites.

Some downsides have been raised against defining a TCP Option to reveal a host identity:

- Conveying an IP address in a TCP Option may be seen as a violation of OSI layers but since IP addresses are already used for the checksum computation, this is not seen as a blocking point. Moreover, updated version of [I-D.wing-nat-reveal-option] does not allow anymore to convey an IP address (the HOST_ID is encoded in 16bits).

- TCP Option space is limited, and might be consumed by the TCP client. Earlier versions of [I-D.wing-nat-reveal-option] discuss two approaches to sending the HOST_ID: sending the HOST_ID in the TCP SYN (which consumes more bytes in the TCP header of the TCP SYN) and sending the HOST_ID in a TCP ACK (which consumes only two bytes in the TCP SYN). Content providers may find it more desirable to receive the HOST_ID in the TCP SYN, as that more closely preserves the HOST_ID received in the source IP address as per current practices. It is more complicated to implement sending the HOST_ID in a TCP ACK, as it can introduce MTU issues

if the ACK packet also contains TCP data, or a TCP segment is lost. The latest specification of the HOST_ID TCP Option, documented at [I-D.wing-nat-reveal-option], allows only to enclose the HOST_ID in the TCP SYN packet.

- When there are several NATs in the path, the original HOST_ID may be lost. In such case, the procedure may not be efficient.

- Interference with current usages such as X-Forwarded-For (see Section 3.8) should be elaborated to specify the behavior of servers when both options are used; in particular specify which information to use: the content of the TCP Option or what is conveyed in the application headers.

- When load-balancers or proxies are in the path, this option does not allow to preserve the original source IP address and source port. Preserving such information is required for logging purposes for instance (e.g., [RFC6302]) . [I-D.abdo-hostid-tcpopt-implementation] defines a TCP Option which allows to reveal various combinations of source information (e.g., source port, source port and source IP address, source IPv6 prefix, etc.).

More discussion about issues raised when extending TCP can be found at [ExtendTCP].

3.6. PROXY Protocol

3.6.1. Description

The solution, referred to as Proxy Protocol [Proxy], does not require any application-specific knowledge. The rationale behind this solution is to prepend each connection with a line reporting the characteristics of the other side's connection as shown in the example depicted in Figure 2:

```
PROXY TCP4 192.0.2.1 192.0.2.15 56324 443\r\n
```

Figure 2: Example of PROXY conection report

Upon receipt of a message conveying this line, the server removes the line. The line is parsed to retrieve the transported protocol. The content of this line is recorded in logs and used to enforce policies.

3.6.2. Analysis

This solution can be deployed in a controlled environment but it can not be deployed to all access services available in the Internet. If the remote server does not support the Proxy Protocol, the session will fail. Other complications will raise due to the presence of firewalls for instance.

As a consequence, this solution is broken and can not be recommended.

3.7. Host Identity Protocol (HIP)

3.7.1. Description

[RFC5201] specifies an architecture which introduces a new namespace to convey an identity information.

3.7.2. Analysis

This solution requires both the client and the server to support HIP [RFC5201]. Additional architectural considerations are to be taken into account such as the key exchanges, etc.

If the address sharing function is required to act as a UDP/TCP-HIP relay, this is not a viable option.

3.8. Inject Application Headers

3.8.1. Description

Another option is to not require any change at the transport nor the IP levels but to convey at the application payload the required information which will be used to disambiguate hosts. This format and the related semantics depend on its application (e.g., HTTP, SIP, SMTP, etc.).

For HTTP, the X-Forwarded-For (XFF) or Forwarded-For ([I-D.ietf-appsawg-http-forwarded]) headers can be used to display the original IP address when an address sharing device is involved. Service Providers operating address sharing devices can enable the feature of injecting the XFF header which will enclose the original IPv4 address or the IPv6 prefix part (see the example shown in Figure 3). The address sharing device has to strip all included XFF headers before injecting their own. Servers may rely on the contents of this field to enforce some policies such as blacklisting misbehaving users. Note that XFF can also be logged by some servers (this is for instance supported by Apache).

```
Forwarded: for=192.0.2.1,for=[2001:db8::1]
Forwarded: proto=https;by=192.0.2.15
```

Figure 3: Example of Forwarded-For

3.8.2. Analysis

Not all applications impacted by the address sharing can support the ability to disclose the original IP address. Only a subset of protocols (e.g., HTTP) can rely on this solution.

For the HTTP case, to prevent users injecting invalid HOST_IDs, an initiative has been launched to maintain a list of trusted ISPs using XFF: See for example the list available at: [Trusted_ISPs] of trusted ISPs as maintained by Wikipedia. If an address sharing device is on the trusted XFF ISPs list, users editing Wikipedia located behind the address sharing device will appear to be editing from their "original" IP address and not from the NATed IP address. If an offending activity is detected, individual hosts can be blacklisted instead of all hosts sharing the same IP address.

XFF header injection is a common practice of load balancers. When a load balancer is in the path, the original content of any included XFF header should not be stripped. Otherwise the information about the "origin" IP address will be lost.

When several address sharing devices are crossed, XFF header can convey the list of IP addresses (e.g., Figure 3). The origin HOST_ID can be exposed to the target server.

XFF also introduces some implementation complexity if the HTTP packet is at or close to the MTU size.

It has been reported that some "poor" implementation may encounter some parsing issues when injecting XFF header.

For encrypted HTTP traffic, injecting XFF header may be broken.

4. HOST_ID and Privacy

IP address sharing is motivated by a number of different factors. For years, many network operators have conserved the use of public IPv4 addresses by making use of Customer Premises Equipment (CPE) that assigns a single public IPv4 address to all hosts within the customer's local area network and uses NAT [RFC3022] to translate between locally unique private IPv4 addresses and the CPE's public address. With the exhaustion of IPv4 address space, address sharing

between customers on a much larger scale is likely to become much more prevalent. While many individual users are unaware of and uninvolved in decisions about whether their unique IPv4 addresses get revealed when they send data via IP, some users realize privacy benefits associated with IP address sharing, and some may even take steps to ensure that NAT functionality sits between them and the public Internet. IP address sharing makes the actions of all users behind the NAT function unattributable to any single host, creating room for abuse but also providing some identity protection for non-abusive users who wish to transmit data with reduced risk of being uniquely identified.

The proposals considered in this document add a measure of uniqueness back to hosts that share a public IP address. The extent of that uniqueness depends on which information is included in the HOST_ID.

The volatility of the HOST_ID information is similar to the source IP address: a distinct HOST_ID may be used by the address sharing function when the host reboots or gets a new internal IP address. As with persistent IP addresses, persistent HOST_IDs facilitate user tracking over time.

As a general matter, the HOST_ID proposals do not seek to make hosts any more identifiable than they would be if they were using a public, non-shared IP address. However, depending on the solution proposal, the addition of HOST_ID information may allow a device to be fingerprinted more easily than it otherwise would be. Should multiple solutions be combined (e.g., TCP Option and XFF) that include different pieces of information in the HOST_ID, fingerprinting may become even easier.

The trust placed in the information conveyed in the HOST_ID is likely to be the same as for current practices with source IP addresses. In that sense, a HOST_ID can be spoofed as this is also the case for spoofing an IP address. Furthermore, users of network-based anonymity services (like Tor) may be capable of stripping HOST_ID information before it reaches its destination.

For more discussion about privacy, refer to [RFC6462].

5. IANA Considerations

This document does not require any action from IANA.

6. Security Considerations

The same security concerns apply for the injection of an IP option, TCP Option and application-related content (e.g., XFF) by the address sharing device. If the server trusts the content of the HOST_ID field, a third party user can be impacted by a misbehaving user to reveal a "faked" original IP address.

7. Acknowledgments

Many thanks to D. Wing and C. Jacquenet for their review, comments and inputs.

Thanks also to P. McCann, T. Tsou, Z. Dong, B. Briscoe, T. Taylor, M. Blanchet, D. Wing and A. Yourtchenko for the discussions in Prague.

Some of the issues related to defining a new TCP Option have been raised by L. Eggert.

Privacy text is provided by A. Cooper.

8. References

8.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

8.2. Informative References

[Count] "A technique for counting NATted hosts", <http://www.cs.columbia.edu/~smb/papers/fnat.pdf>.

[ExtendTCP] Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A.,

Handley, M. and H. Tokuda,, "Is it still possible to extend TCP?", November 2011, <http://nrg.cs.ucl.ac.uk/mjh/tmp/mboxes.pdf>.

[I-D.abdo-hostid-tcpopt-implementation]
Abdo, E., Boucadair, M., and J. Queiroz, "HOST_ID TCP Options: Implementation & Preliminary Test Results", draft-abdo-hostid-tcpopt-implementation-02 (work in progress), January 2012.

[I-D.chen-intarea-v4-uid-header-option]
Wu, Y., Ji, H., Chen, Q., and T. ZOU), "IPv4 Header Option For User Identification In CGN Scenario", draft-chen-intarea-v4-uid-header-option-00 (work in progress), March 2011.

[I-D.ietf-appsawg-http-forwarded]
Petersson, A. and M. Nilsson, "Forwarded HTTP Extension", draft-ietf-appsawg-http-forwarded-00 (work in progress), January 2012.

[I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", draft-ietf-behave-lsn-requirements-05 (work in progress), November 2011.

[I-D.ietf-intarea-ipv4-id-update]
Touch, J., "Updated Specification of the IPv4 ID Field", draft-ietf-intarea-ipv4-id-update-04 (work in progress), September 2011.

[I-D.wing-nat-reveal-option]
Yourtchenko, A. and D. Wing, "Revealing hosts sharing an IP address using TCP option", draft-wing-nat-reveal-option-03 (work in progress), December 2011.

[I-D.yourtchenko-nat-reveal-ping]
Yourtchenko, A., "Revealing hosts sharing an IP address using ICMP Echo Request", draft-yourtchenko-nat-reveal-ping-00 (work in progress), March 2012.

[Not_An_Option]
R. Fonseca, G. Porter, R. Katz, S. Shenker, and I. Stoica,, "IP options are not an option", 2005, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/

EECS-2005-24.html>.

[Options] Alberto Medina, Mark Allman, Sally Floyd, "Measuring Interactions Between Transport Protocols and Middleboxes", 2005, <http://conferences.sigcomm.org/imc/2004/papers/p336-medina.pdf>.

[Proxy] Tarreau, W., "The PROXY protocol", November 2010, <http://haproxy.1wt.eu/download/1.5/doc/proxy-protocol.txt>.

[RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, June 2011.

[RFC6462] Cooper, A., "Report from the Internet Privacy Workshop", RFC 6462, January 2012.

[Trusted_ISPs] "Trusted XFF list", <http://meta.wikimedia.org/wiki/XFF_project#Trusted_XFF_list>.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Joe Touch
USC/ISI

Email: touch@isi.edu

Pierre Levis
France Telecom
Caen, 14000
France

Email: pierre.levis@orange.com

Reinaldo Penno
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, California 94089
USA

Email: rpenno@juniper.net